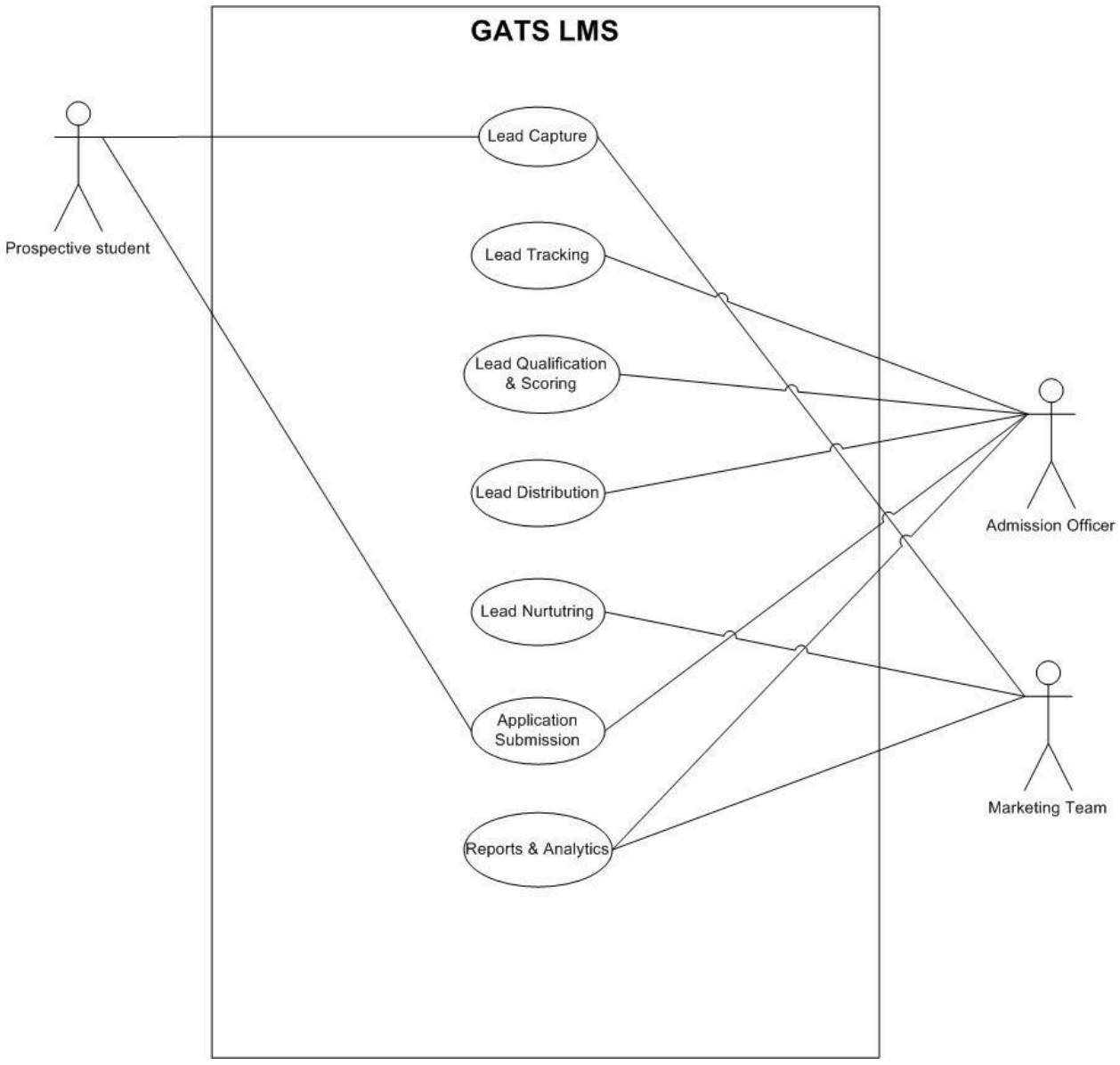
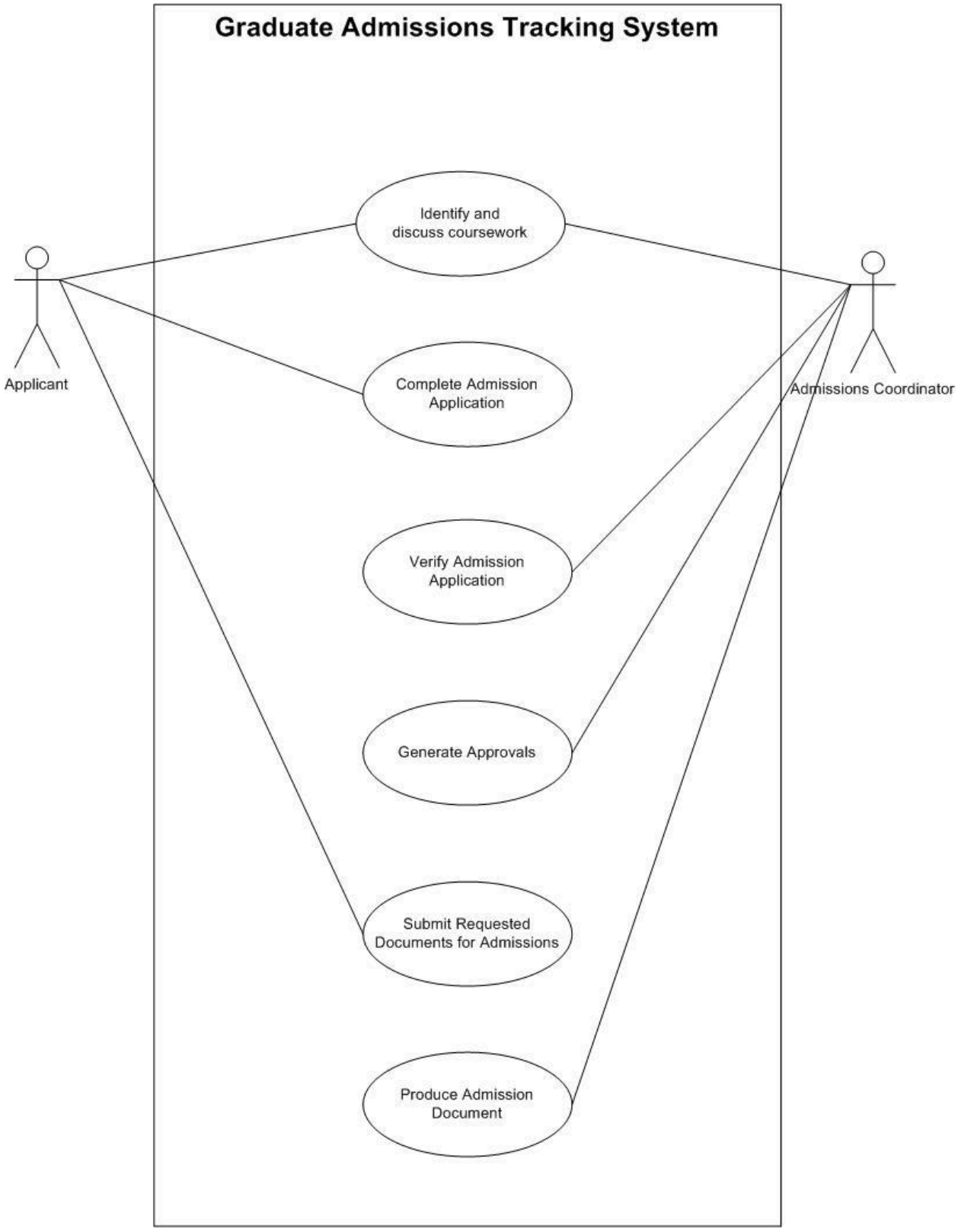
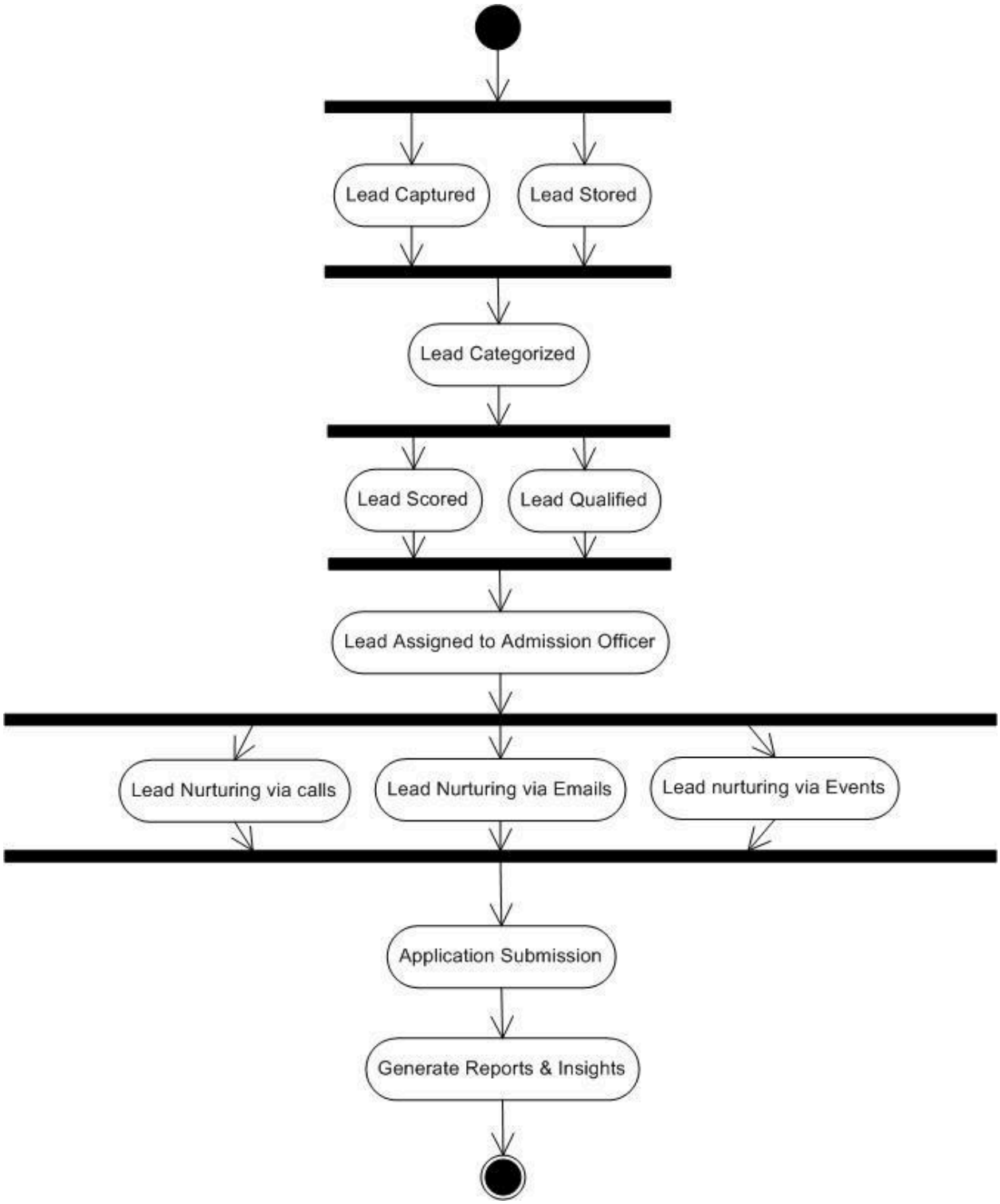
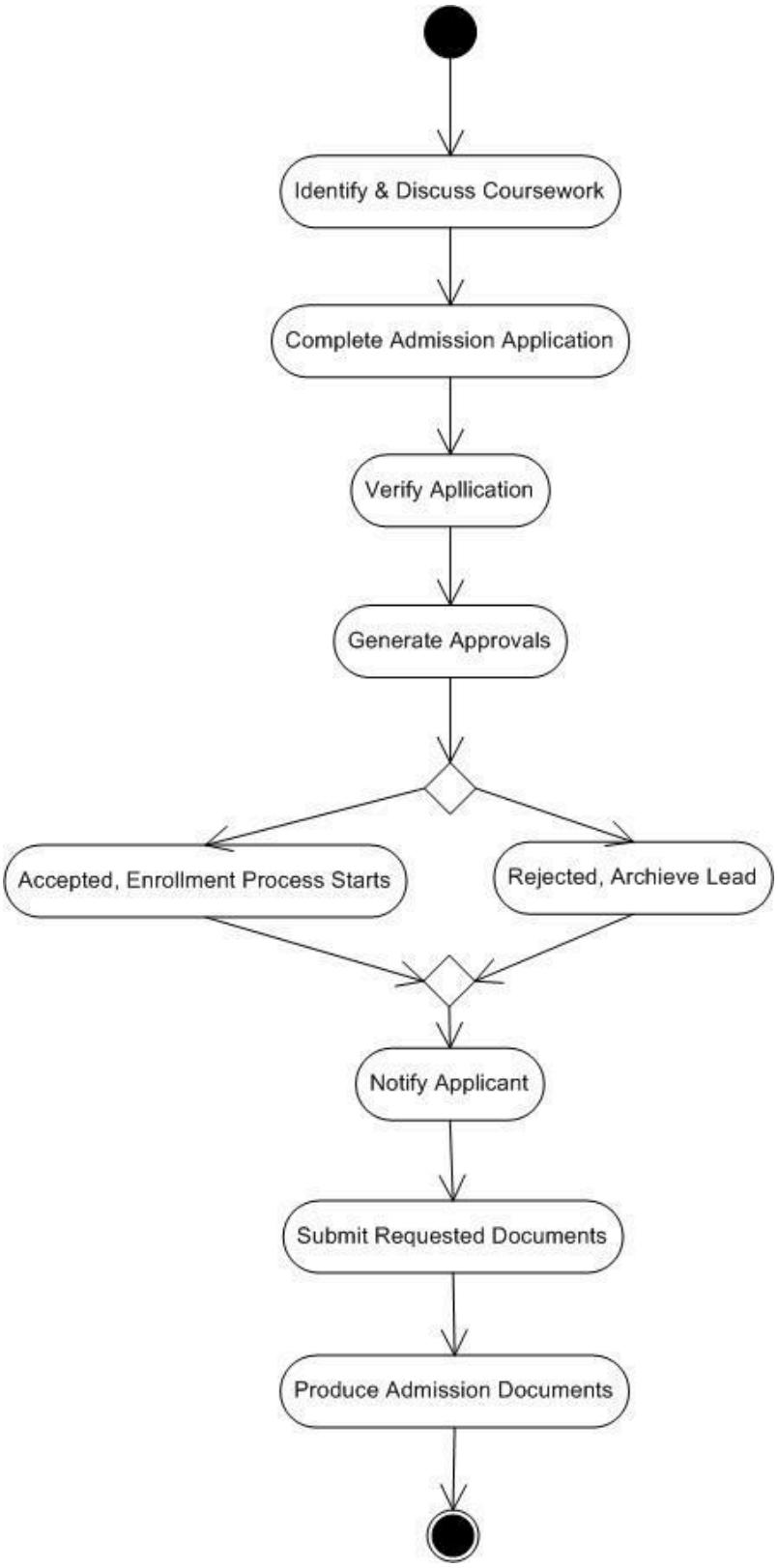
GATS-2

Document 6: Use Case Diagram, Activity Diagram, and Use Specification Document




Graduate Admissions Tracking System
Identify and discuss coursework
Complete Admission Application
Verify Admission Application
Generate Approvals
Submit Requested Documents for Admissions
Produce Admission Document
Applicant
Admissions Coordinator


Lead Captured
Lead Stored
Lead Categorized
Lead Scored
Lead Qualified
Lead Assigned to Admission Officer
Lead Nurturing via calls
Lead Nurturing via Emails
Lead nurturing via Events
Application Submission
Generate Reports & Insights


Identify & Discuss Coursework
Complete Admission Application
Verify Apllication
Generate Approvals
Accepted, Enrollment Process Starts
Rejected, Archieve Lead
Notify Applicant
Submit Requested Documents
Produce Admission Documents

Use Case Specifications for GATS LMS

Use Case ID:	UC-001
Use Case Name:	Lead Capture
Use Case Description:	This use case describes how prospective students submit their details through various channels, and the system captures the lead.
Actors:	● **Primary Actor:** Prospective Student ● **Secondary Actor:** Marketing Team
Basic Flow:	1. The prospective student submits their details via an online form, chatbot, or phone inquiry. 2. The system captures the details and stores them in the database. 3. The marketing team reviews and validates the lead. 4. The system assigns a unique Lead ID. 5. A confirmation email/SMS is sent to the student.
Alternate Flow:	● If a student submits incomplete details, the system prompts for missing fields.
Exceptional Flows:	● A failure message is displayed if the lead submission fails due to a system error, and the student is advised to retry later.
Pre-Conditions:	● The system must be online and operational.
Post-Conditions:	● Lead details are successfully stored.
Assumptions:	● The student provides correct and accurate information.
Constraints:	● The system must handle high lead volume efficiently.
Dependencies:	● Internet connectivity and form submission APIs.
Inputs and Outputs:	● **Input:** Student Name, Email, Phone, Program of Interest, etc. ● **Output:** Confirmation message, Unique Lead ID.
Business Rules:	● Each lead must have a unique identifier.

Use Case ID:	UC-001
Miscellaneous Information:	● Data privacy regulations must be followed.

Use Case ID:	UC-002
Use Case Name:	Lead Tracking
Use Case Description:	This use case describes how an admissions officer or marketing team tracks the status of a lead in the system.
Actors:	● **Primary Actor:** Admissions Officer ● **Secondary Actor:** Marketing Team
Basic Flow:	1. The user logs into the system. 2. They search for a lead by name, email, or Lead ID. 3. The system displays the lead details and current status. 4. The user updates the lead status if required. 5. The system logs all changes for audit purposes.
Alternate Flow:	● The system provides a "No record found" message if no lead is found.
Exceptional Flows:	● An appropriate error message is displayed, if the system encounters a database error.
Pre-Conditions:	● The user must have login credentials.
Post-Conditions:	● Lead details are retrieved and displayed.
Assumptions:	● The lead data is correctly entered into the system.
Constraints:	● The system should support multiple concurrent searches.
Dependencies:	● Database availability.
Inputs and Outputs:	● **Input:** Lead ID, Name, Email ● **Output:** Lead Details, Status
Business Rules:	● Only authorized personnel can update lead status.
Miscellaneous Information:	● System logs should be maintained for future reference.

Use Case ID:	UC-003
Use Case Name:	Lead Qualification
Use Case Description:	Determines if a lead meets the university's admission criteria.
Actors:	**Primary Actor:** Admissions Officer
Basic Flow:	1. The system scores the lead based on predefined criteria. 2. Admissions Officer reviews and qualifies the lead. 3. Qualified leads move to the next stage.
Alternate Flow:	● The system archives a lead, if it does not meet the criteria.
Exceptional Flow:	● System error prevents lead scoring.
Pre-Conditions:	● Scoring rules must be defined in the system.
Post-Conditions:	● Leads are categorized as qualified or unqualified.
Assumptions:	● The lead scoring model is correctly configured and updated. ● The admissions officer follows the system's recommendations for qualification. ● Leads provide accurate information during the initial capture phase.
Constraints:	● Lead scoring criteria must align with university admission policies. ● The system should process large numbers of leads efficiently. ● The scoring mechanism must comply with legal and ethical standards.
Dependencies:	● Lead Capture (UC-001) must be completed before qualification. ● System access for the admissions officer. ● Availability of scoring algorithms and historical data for evaluation.
Inputs and Outputs:	● **Inputs**: Lead details (academic background, test scores, program of interest, location). ● **Outputs**: Lead qualification status (Qualified/Unqualified), next steps (Application submission, archive).
Business Rules:	● A lead must meet at least 70% of the university's admission criteria to be marked as "Qualified."

	● If a lead is unqualified, it should be archived but remain accessible for future re-evaluation. ● Only authorized personnel can override system recommendations.
Miscellaneous Information:	● The lead scoring algorithm should be reviewed periodically to adapt to changes in admission policies. ● The system should notify unqualified leads politely, providing alternative options (e.g., preparatory courses, scholarships).

Use Case ID:	UC-004
Use Case Name:	Lead Distribution
Use Case Description:	Assigns leads to different departments.
Actors:	**Primary Actor:** System **Secondary Actor:** Admissions Officer
Basic Flow:	1. Leads are categorized by program interest. 2. Leads are assigned to relevant department staff.
Alternate Flow:	● If a specific department is overloaded with leads, the system redistributes them to other available staff. ● If no relevant department is found, the lead remains unassigned and flagged for manual review.
Exceptional Flow:	● If the system fails to assign a lead due to a server issue, an error message is logged, and the lead remains in the queue for reassignment. ● If a lead is mistakenly assigned to the wrong department, an admissions officer can manually reassign it.
Pre-Conditions:	● The system must have categorized leads based on interest and program selection. ● Departments and staff must be registered in the system.
Post-Conditions:	● Leads are successfully assigned to the relevant department or staff. ● If assignment fails, leads remain in an unassigned queue for manual intervention.

Assumptions:	● The system has updated and accurate information on department staff availability. ● Leads provide correct program preferences during lead capture. ● The distribution algorithm follows pre-configured rules.
Constraints:	● The system should prevent duplicate assignments. ● The assignment algorithm must be scalable to handle a high volume of leads. ● Manual reassignment should only be allowed for authorized personnel.
Dependencies:	● Lead Capture (UC-001) and Lead Qualification (UC-003) must be completed before distribution. ● System integration with CRM tools for lead assignment. ● Availability of admissions officers for lead follow-ups.
Inputs and Outputs:	● **Input:** Qualified leads with program preferences, department staff availability. ● **Output:** Assigned leads, notifications to assigned staff, logs of distribution.
Business Rules:	● Leads must be assigned within **24 hours** of qualification. ● Priority leads (e.g., high-scoring or scholarship applicants) must be assigned first. ● If a lead remains unassigned for **48 hours**, an alert is sent to the admissions manager.
Miscellaneous Information:	● Lead distribution should be balanced to avoid overloading specific departments. ● The system should allow batch assignments for efficiency. ● A reassignment log should be maintained for audit purposes.

Use Case ID:	UC-005
Use Case Name:	Lead Nurturing
Use Case Description:	Engages with leads via emails, calls, and events.
Actors:	● **Primary Actor:** Marketing Team
Basic Flow:	1. The system sends automated emails to leads. 2. Marketing representatives follow up with phone calls. 3. The system tracks engagement metrics.

Alternate Flow:	● If a lead does not respond to initial email follow-ups, the system escalates engagement efforts (e.g., personalized email, SMS, or call). ● Leads who engage but need more time can be placed in a follow-up sequence for a later campaign.
Exceptional Flow:	● If an email bounces, the system flags the lead for a manual contact attempt. ● If a lead unsubscribes, the system removes them from future communications while maintaining a record.
Pre-Conditions:	● The system must have valid contact details for the lead. ● The lead must not have opted out of communication. ● Marketing content (emails, messages) must be pre-approved.
Post-Conditions:	● Leads receive nurturing content through emails or calls. ● Engagement data is recorded in the system. ● Leads may move to the application submission stage.
Assumptions:	● Leads are interested in university programs and will engage with the outreach efforts. ● The marketing team will follow up with high-priority leads. ● Automated email and SMS systems are functional.
Constraints:	● The system must comply with email marketing regulations (e.g., GDPR, CAN-SPAM). ● Limited capacity for manual follow-ups, requiring prioritization. ● Email frequency should not exceed a predefined threshold to prevent spam classification.
Dependencies:	● Lead Capture (UC-001) and Lead Qualification (UC-003) must be completed before nurturing. ● Integration with CRM tools and email automation platforms. ● Availability of marketing representatives for manual follow-ups.
Inputs and Outputs:	● **Inputs:** Lead contact details, program interests, past engagement history. ● **Outputs:** Emails, SMS, call logs, engagement metrics (open rates, click rates, response rates).
Business Rules:	● Leads should receive a maximum of 3 emails per week unless they respond or engage.

	● Leads showing high engagement should be prioritized for direct follow-up. ● If a lead remains unresponsive for 60 days, they are marked as inactive.
Miscellaneous Information:	● The system should allow personalization of messages for better engagement. ● Follow-up strategies should be adjusted based on lead behavior (e.g., interest in a particular program). ● Nurturing campaigns should be periodically reviewed and optimized based on effectiveness.

Use Case ID:	UC-006
Use Case Name:	Application Submission
Use Case Description:	Allows qualified leads to submit applications.
Actors:	● Primary Actor: Prospective Student
Basic Flow:	1. The student completes the application form. 2. The system validates and submits the form.
Alternate Flow:	● If the student does not complete the application in one session, the system saves the progress for later completion. ● If the student selects an unavailable program, the system suggests alternative programs.
Exceptional Flow:	● If the system encounters a validation error (e.g., missing required fields), an error message is displayed, and the student is prompted to correct it. ● If the document upload fails, the system prompts the user to retry.
Pre-Conditions:	● The student must be a qualified lead and have login credentials. ● Application forms and fee payment options must be available.
Post-Conditions:	● The application is successfully submitted and stored in the system. ● The student receives a confirmation email.

Assumptions:	● The student has access to all necessary documents for submission. ● The system is operational and can process applications in real-time.
Constraints:	● Application submission must comply with university deadlines. ● The system should support high traffic during peak admission periods.
Dependencies:	● Lead Qualification (UC-003) must be completed before application submission. ● Integration with the document verification system. ● Availability of payment gateways for application fees.
Inputs and Outputs:	● **Inputs**: Student details, academic records, application form, required documents, payment details. ● **Outputs**: Application confirmation, application tracking ID, payment receipt.
Business Rules:	● All required fields must be filled before submission. ● Only qualified leads can submit applications. ● Application fee payment must be verified before processing.
Miscellaneous Information:	● The system should allow students to track their application status post-submission. ● Support should be available for students facing issues during submission.

Use Case ID:Actors:	UC-007
Use Case Name:	Application Review
Use Case Description:	Evaluates submitted applications.
Actors:	● **Primary Actor:** Admissions Officer
Basic Flow:	1. The admissions officer reviews the application. 2. The system logs decisions (Accepted/Rejected).
Alternate Flow:	● If additional documents are required, the system notifies the applicant to submit them before proceeding. ● If the application requires a faculty review, it is forwarded to

	the appropriate reviewer before a final decision.
Exceptional Flow:	● If the system encounters an error retrieving the application, an error message is displayed, and the admissions officer can retry. ● If an application is accidentally reviewed under the wrong category, the admissions officer can request reassessment.
Pre-Conditions:	● The application must be submitted and available for review. ● Admissions officers must have access to the system with appropriate permissions.
Post-Conditions:	● The application is marked as "Accepted," "Rejected," or "Pending Additional Review." ● The applicant is notified of the decision.
Assumptions:	● All necessary documents are available for review. ● Admissions officers follow university guidelines while making decisions.
Constraints:	● The review process must be completed within the university's application deadline. ● The system should handle multiple concurrent application reviews.
Dependencies:	● Application Submission (UC-006) must be completed before the review. ● Integration with document verification and academic evaluation tools.
Inputs and Outputs:	● **Inputs:** Submitted application, uploaded documents, applicant information. ● **Outputs:** Decision status (Accepted/Rejected/Pending), feedback notes, automated applicant notifications.
Business Rules:	● Applications must be reviewed within **10 business days** of submission. ● Only authorized admissions officers can modify application statuses. ● If an application is incomplete, it cannot be reviewed.
Miscellaneous Information:	● The system should maintain an audit trail of all reviews and decisions. ● Applicants should have a way to track the progress of their application review.

Use Case ID:	UC-008
Use Case Name:	Lead Conversion
Use Case Description:	Converts a lead into an enrolled student.
Actors:	● **Primary Actor:** System
Basic Flow:	The system updates the lead status to "Enrolled."
Alternate Flow:	● If a student does not confirm enrollment immediately, the system schedules a follow-up reminder. ● If the student chooses to defer admission, the system updates the record accordingly.
Exceptional Flow:	● If the student fails to pay the enrollment fee, the system places the lead in a "Pending Payment" status. ● If the lead decides not to enroll, the system updates the status to "Declined" and removes them from further follow-ups.
Pre-Conditions:	● The student's application must be approved. ● The system must have an active enrollment process in place.
Post-Conditions:	● The lead is converted into an enrolled student. ● The system updates the student database and generates a student ID.
Assumptions:	● The student intends to proceed with enrollment after application approval. ● Payment gateways and enrollment processing systems are functional.
Constraints:	● The conversion process must be completed within the university's enrollment deadline. ● The system should handle multiple conversions simultaneously without performance issues.
Dependencies:	● Application Review (UC-007) must be completed before conversion. ● Integration with the student management system.
Inputs and Outputs:	● **Inputs:** Approved application, payment confirmation, student details. ● **Outputs:** Enrollment confirmation, student ID, welcome

	email.
Business Rules:	● Only approved applications can proceed to enrollment. ● Enrollment fees must be paid before a lead is marked as "Enrolled." ● If a student defers enrollment, their record should be retained for future admission.
Miscellaneous Information:	● The system should allow admissions officers to manually adjust lead conversion statuses if needed. ● Automated communication should be sent to new students with next steps after enrollment.

Use Case ID:	UC-009
Use Case Name:	Reporting & Analytics
Use Case Description:	Generates insights on lead performance.
Actors:	● **Primary Actor:** Administrator
Basic Flow:	1. The administrator selects report criteria. 2. The system generates reports.
Alternate Flow:	● If the selected report type does not exist, the system suggests alternative report types. ● If a user requests a customized report, the system allows additional filtering criteria.
Exceptional Flow:	● If the system fails to generate a report due to a server error, an error message is displayed. ● If data is incomplete or missing, the system notifies the administrator.
Pre-Conditions:	● The user must have the necessary permissions to generate reports. ● The system must have sufficient data to generate meaningful analytics.
Post-Conditions:	● The requested report is generated and displayed. ● The user can download or print the report.
Assumptions:	● The data in the system is up-to-date and accurate. ● The user knows how to interpret the generated reports.

Constraints:	● Large reports may take longer to generate. ● The system must comply with data security and privacy policies.
Dependencies:	● Lead tracking (UC-002) and other data collection processes must be functioning correctly. ● Database availability and performance.
Inputs and Outputs:	● **Inputs:** Date range, report type, filters (e.g., department, lead source). ● **Outputs:** Lead performance reports, conversion analytics, application trends.
Business Rules:	● Only authorized personnel can access detailed analytics. ● Reports must be stored securely and comply with university policies
Miscellaneous Information:	● The system should allow exporting reports in multiple formats (PDF, Excel, etc.). ● The system may provide graphical data representations like charts and graphs.

Use Case ID:	UC-010
Use Case Name:	System Notifications
Use Case Description:	Sends automated notifications to leads and staff.
Actors:	● **Primary Actor:** System
Basic Flow:	The system triggers notifications based on lead status updates.
Alternate Flow:	● If the recipient's email or phone number is invalid, the system logs the failure and retries sending after a set interval. ● The system skips sending messages if the recipient has opted out of notifications.
Exceptional Flow:	● If the notification service is down, the system queues the messages and retries later. ● An alert is sent to the administrator if the system fails to send multiple notifications.
Pre-Conditions:	● The system must have an active notification service. ● The recipient's contact details must be valid.

Post-Conditions:	● The recipient receives the notification successfully. ● The system logs the notification details for audit purposes.
Assumptions:	● Users have provided valid contact information. ● The system is configured to send notifications based on predefined rules.
Constraints:	● Notifications should be sent within a reasonable time frame (e.g., within 5 minutes of a status update). ● The system should comply with data privacy regulations regarding messaging and communication.
Dependencies:	● Integration with email/SMS service providers. ● Lead tracking and application management modules must trigger notifications.
Inputs and Outputs:	● **Inputs:** Event triggers (e.g., application status update, payment confirmation), recipient details. ● **Outputs:** Email/SMS/App notifications, confirmation logs.
Business Rules:	● Notifications should only be sent to authorized recipients. ● Sensitive information (e.g., payment details) should not be included in plain text. ● Users should have the option to opt in/out of non-critical notifications.
Miscellaneous Information:	● The system should allow administrators to configure notification templates. ● The system may provide a dashboard for tracking notification delivery success rates.

Use Case Specifications for Graduate Admissions Tracking System (GATS)

Use Case ID	**UC001**
Use Case Name	Login
Created by	Business Analyst
Use Case Description	This use case allows users (applicants, admissions officers, and admins) to securely log in to the Graduate Admissions Tracking System.

Actors	● **Primary Actor:** User (Applicant, Admissions Officer, Admin) ● **Secondary Actor:** System, Database
Basic Flow	1. User navigates to the login page. 2. User enters their username and password. 3. System validates the credentials. 4. If valid, the system logs in the user and redirects them to their dashboard.
Alternate Flow	● **Forgot Password:** 1. User clicks "Forgot Password." 2. System prompts the user to enter their email. 3. System sends a password reset link to the email.
Exceptional Flow	● **Invalid Credentials:** 1. System displays an error message for invalid credentials. 2. User is prompted to re-enter credentials.
Pre-conditions	● User must be registered in the system.
Post-conditions:	● User gains access to their dashboard.
Assumptions	● User has an active account.
Constraints	● he login process must be completed within 60 seconds of inactivity.
Dependencies	● System must connect to the database for user verification.
Inputs and Outputs	● **Inputs:** Username, password. ● **Outputs:** Dashboard access or error message.
Business Rules	● User must provide correct credentials.
Miscellaneous Information	● The system supports multi-factor authentication in future updates.

Use Case ID	**UC002**
Use Case Name	Submit Application
Created by	Business Analyst
Use Case Description	Applicants can submit their graduate application online, including

	providing personal details and uploading documents.
Actors	● **Primary Actor:** Applicant ● **Secondary Actor:** System, Database
Basic Flow	1. Applicant logs in to the system. 2. Applicant navigates to the "Submit Application" section. 3. Applicant fills in the required details. 4. Applicant uploads supporting documents. 5. Applicant clicks "Submit." 6. System validates the information and confirms the submission.
Alternate Flow	● **Incomplete Submission:** 1. System highlights missing or incorrect fields. 2. Applicant corrects errors and re-submits.
Exceptional Flow	● **File Upload Error:** 1. System rejects unsupported file types. 2. Applicant is prompted to upload files in the correct format.
Pre-conditions	● Applicant must have an active account and be logged in.
Post-conditions	● Application is submitted and stored in the database.
Assumptions	● Applicant has all required documents ready.
Constraints	● File uploads must not exceed 10MB per document.
Dependencies	● System must validate and store the application data.
Inputs and Outputs	● **Inputs:** Applicant's personal details, uploaded documents. ● **Outputs:** Confirmation message.
Business Rules	● All required fields must be completed before submission.
Miscellaneous Information	● The system supports multiple file types for uploads.

Use Case ID	**UC003**
Use Case Name	Track Application Status
Created by	Business Analyst

Use Case Description	This use case allows applicants to track the real-time status of their submitted applications.
Actors	● **Primary Actor:** Applicant ● **Secondary Actor:** System
Basic Flow	1. Applicant logs in to the system. 2. Applicant navigates to the "Application Status" page. 3. System displays the current status of the application (e.g., submitted, under review, accepted).
Alternate Flow	● **Multiple Applications:** 1. Applicant selects a specific application from a list. 2. System displays the status of the selected application.
Exceptional Flow	● **System Error:** 1. If the system cannot retrieve the status, an error message is displayed. 2. Applicant is advised to try again later.
Pre-conditions	● Applicant must have submitted at least one application.
Post-conditions	● Application status is displayed to the applicant.
Assumptions	● All applications have updated statuses in the database.
Constraints	● The system must handle concurrent status requests efficiently.
Dependencies	● Application data must be updated in the database.
Inputs and Outputs	● **Inputs:** Application ID or selection. ● **Outputs:** Application status.
Business Rules	● Status updates must be accurate and reflect the latest processing stage.
Miscellaneous Information	● The system allows viewing of historical statuses if needed.

Use Case ID	**UC004**
Use Case Name	Approve or Reject Applications

Created by	Business Analyst
Use Case Description	Admissions officers review and make decisions on submitted applications based on predefined criteria.
Actors:	● **Primary Actor:** Admissions Officer ● **Secondary Actor:** System
Basic Flow:	1. Admissions officer logs in to the system. 2. Officer navigates to the "Application Review" section. 3. Officer selects an application for review. 4. Officer reviews the applicant's details and documents. 5. Officer approves or rejects the application. 6. System updates the application status accordingly.
Alternate Flow:	● **Request Additional Information:** 1. Officer requests additional details or documents from the applicant. 2. System notifies the applicant about the request.
Exceptional Flow:	● **System Timeout:** 1. If the system times out during the review, progress is saved automatically. 2. Officer resumes the review upon re-login.
Pre-conditions:	● The application must be submitted and verified.
Post-conditions:	● Application status is updated (approved, rejected, or pending additional details).
Assumptions:	● Officers have access to complete applicant information.
Constraints:	● The review process must be completed within predefined deadlines.
Dependencies:	● System must provide accurate and complete application details.
Inputs and Outputs:	● **Inputs:** Application details, documents. ● **Outputs:** Updated application status.
Business Rules:	● Applications must meet eligibility criteria for approval.
Miscellaneous Information:	● The system tracks review progress and timestamps decisions.

Use Case ID:	**UC005**
Use Case Name:	Document Upload
Created by:	Business Analyst
Use Case Description:	This use case allows applicants to upload supporting documents for their applications.
Actors:	● **Primary Actor:** Applicant ● **Secondary Actor:** System
Basic Flow:	1. Applicant navigates to the "Upload Documents" section. 2. Applicant selects the document type and file from their device. 3. System verifies the file format and size. 4. System uploads and stores the document securely. 5. Applicant receives a confirmation message.
Alternate Flow:	● **Multiple Documents:** 1. Applicant uploads multiple documents in one session. 2. System processes each document sequentially.
Exceptional Flow:	● **Unsupported Format:** 1. System rejects unsupported file formats. 2. Applicant is prompted to upload a valid file.
Pre-conditions:	● Applicant must be logged in.
Post-conditions:	● Documents are securely stored and linked to the application.
Assumptions:	● Applicants have digital copies of their documents.
Constraints:	● File size must not exceed 10MB per document.
Dependencies:	● System must have sufficient storage and processing capacity.
Inputs and Outputs:	● **Inputs:** Document files, document type. ● **Outputs:** Confirmation message, stored documents.
Business Rules:	● Only required document types can be uploaded.
Miscellaneous	● The system supports drag-and-drop functionality for file

Information:	uploads.

Use Case ID:	**UC006**
Use Case Name:	Application Form Validation
Created by:	Business Analyst
Use Case Description:	Validates the application form data to ensure all required fields are completed accurately.
Actors:	● **Primary Actor:** System ● **Secondary Actor:** Applicant
Basic Flow:	1. Applicant submits the application form. 2. System checks for completeness of all required fields. 3. System validates the format and content of the entered data. 4. If validation passes, the application form is marked as "Ready for Review."
Alternate Flow:	● **Incomplete Form:** 1. If required fields are missing, the system highlights the incomplete sections. 2. Applicant is prompted to complete the missing fields before resubmission.
Exceptional Flow:	● **Invalid Data:** 1. If the format of certain fields is incorrect (e.g., invalid email or phone number), the system provides an error message. 2. Applicant must correct the errors and resubmit the form.
Pre-conditions:	● Applicant must be logged in. ● The application form must be accessible.
Post-conditions:	● The application form is validated and marked as "Ready for Review" if all checks are passed.
Assumptions:	● Applicants have the necessary information to complete the form.
Constraints:	● Data validation must comply with institutional standards.
Dependencies:	● The database must be available for data validation checks.

Inputs and Outputs:	● **Inputs:** Application form data. ● **Outputs:** Validation status, error messages (if any).
Business Rules:	● All required fields must be completed before submission. ● Data must adhere to specified formats and constraints.
Miscellaneous Information:	● System supports real-time validation and auto-save functionality during form filling.

Use Case ID:	**UC007**
Use Case Name:	Payment Processing
Created by:	Business Analyst
Use Case Description:	Facilitates secure processing of application fees for applicants.
Actors:	● **Primary Actor:** Applicant ● **Secondary Actor:** Payment Gateway, System
Basic Flow:	1. Applicant navigates to the "Payment" section of the application. 2. Applicant selects a payment method (e.g., credit card, debit card, online banking). 3. System redirects the applicant to the payment gateway. 4. Applicant enters payment details and confirms the transaction. 5. Payment gateway processes the transaction and sends a success or failure response to the system. 6. System updates the application status to "Payment Completed" if the transaction is successful. 7. Applicant receives a payment confirmation message and receipt.
Alternate Flow:	● **Payment Failure:** 1. If the transaction fails, the system notifies the applicant with an error message. 2. Applicant is prompted to retry the payment or use an alternate payment method.
Exceptional Flow:	● **Gateway Timeout:** 1. If the payment gateway times out, the system retries the transaction up to three times. 2. If the retries fail, the system notifies the applicant

	and logs the issue for review.
Pre-conditions:	● Applicant must have completed the application form. ● The payment gateway must be operational.
Post-conditions:	● Payment is successfully processed, and the application status is updated.
Assumptions:	● Applicants have a valid payment method.
Constraints:	● Payment must be completed within the application deadline.
Dependencies:	● Payment gateway service must be integrated and operational.
Inputs and Outputs:	● **Inputs:** Payment details (e.g., card number, expiry date). ● **Outputs:** Payment confirmation message, updated application status.
Business Rules:	● Payments must comply with financial regulations and institutional policies.
Miscellaneous Information:	● The system supports multiple payment methods for user convenience.

Use Case ID:	**UC008**
Use Case Name:	Notify Admission Decision
Created by:	Business Analyst
Use Case Description:	Notify applicants about their admission decisions (e.g., accepted, waitlisted, rejected) through email and system notifications.
Actors:	● **Primary Actor:** System ● **Secondary Actor:** Applicant
Basic Flow:	1. Admission decision is finalized by the admissions officer. 2. System generates a notification for the applicant. 3. Notification is sent to the applicant's registered email and displayed on their dashboard.
Alternate Flow:	● **Notification Failure:** 1. System retries sending the notification if the email

	fails initially.
Exceptional Flow:	● **Invalid Email Address:** 1. System logs the error and flags the applicant's account for manual review.
Pre-conditions:	● Admission decision must be finalized.
Post-conditions:	● Applicant receives their admission decision.
Assumptions:	● Applicant's email address is valid and active.
Constraints:	● Notification must be delivered within 1 hour of decision finalization.
Dependencies:	● System email service must be operational.
Inputs and Outputs:	● **Inputs:** Admission decision. ● **Outputs:** Notification to applicant.
Business Rules:	● Notifications must comply with institutional communication policies.
Miscellaneous Information:	● Notifications include next steps for accepted applicants.

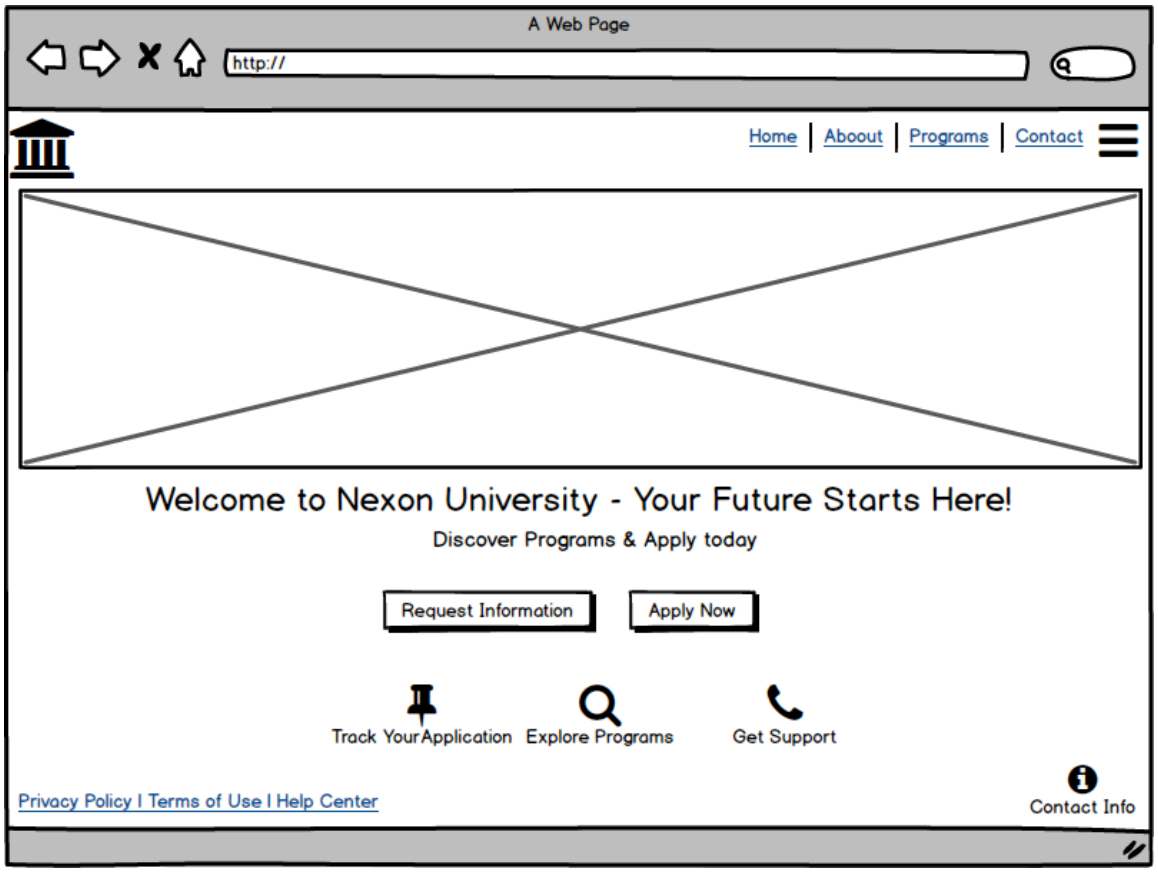
Use Case ID:	**UC009**
Use Case Name:	Account Lockout
Created by:	Business Analyst
Use Case Description:	Enhance security by locking user accounts after multiple unsuccessful login attempts.
Actors:	● **Primary Actor:** System ● **Secondary Actor:** User
Basic Flow:	1. User enters incorrect credentials. 2. System tracks the number of failed attempts. 3. After three failed attempts, the account is locked. 4. System notifies the user of the lockout and provides recovery options.
Alternate Flow:	● **Recovery Process:** 1. User clicks on the account recovery link.

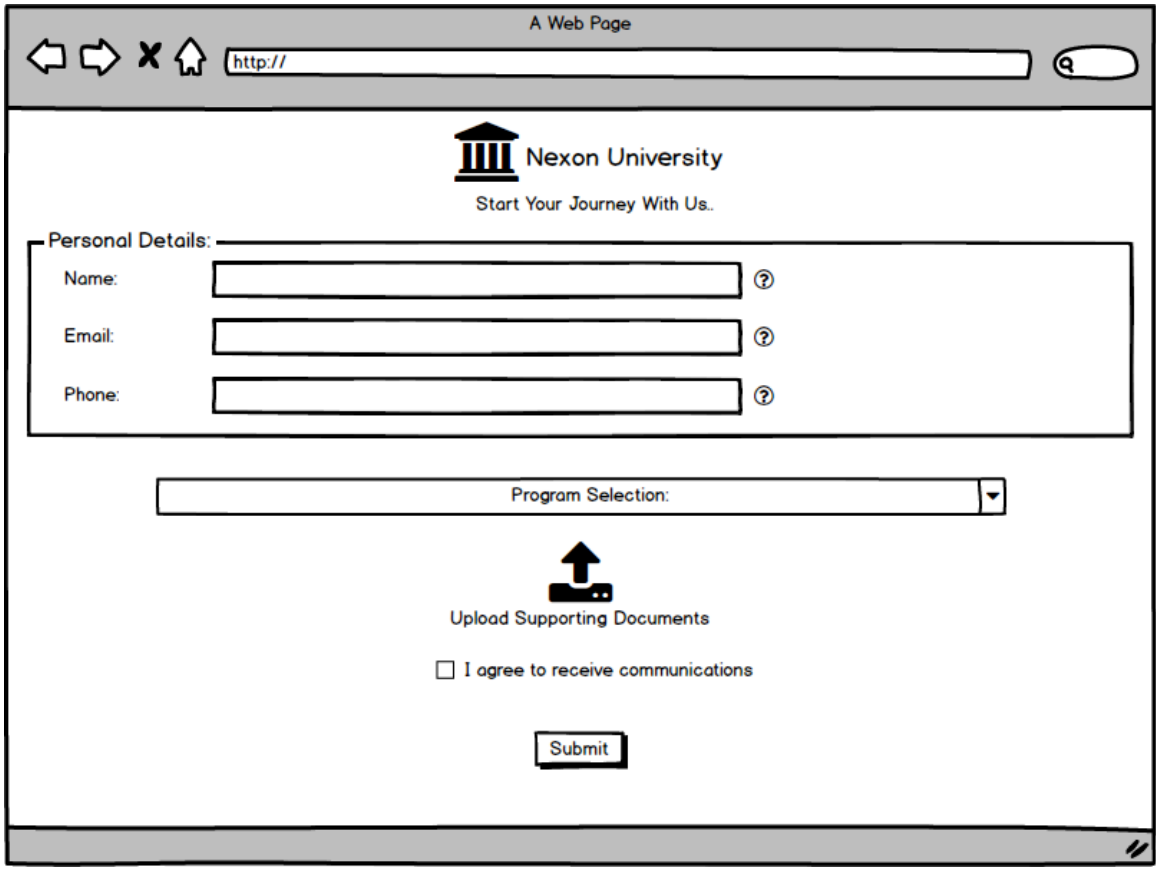
	2. System verifies the user's identity through email or security questions.
Exceptional Flow:	● **Invalid Recovery Attempts:** 1. System blocks further recovery attempts after three invalid attempts.
Pre-conditions:	● User must have an existing account.
Post-conditions:	● Account is locked and flagged for security.
Assumptions:	● Users will not intentionally exceed login attempts.
Constraints:	● Lockout duration is configurable by the admin.
Dependencies:	● System's user authentication service.
Inputs and Outputs:	● **Inputs:** Login credentials. ● **Outputs:** Lockout notification.
Business Rules:	● Lockout rules must comply with security best practices.
Miscellaneous Information:	● Admins can manually unlock accounts.

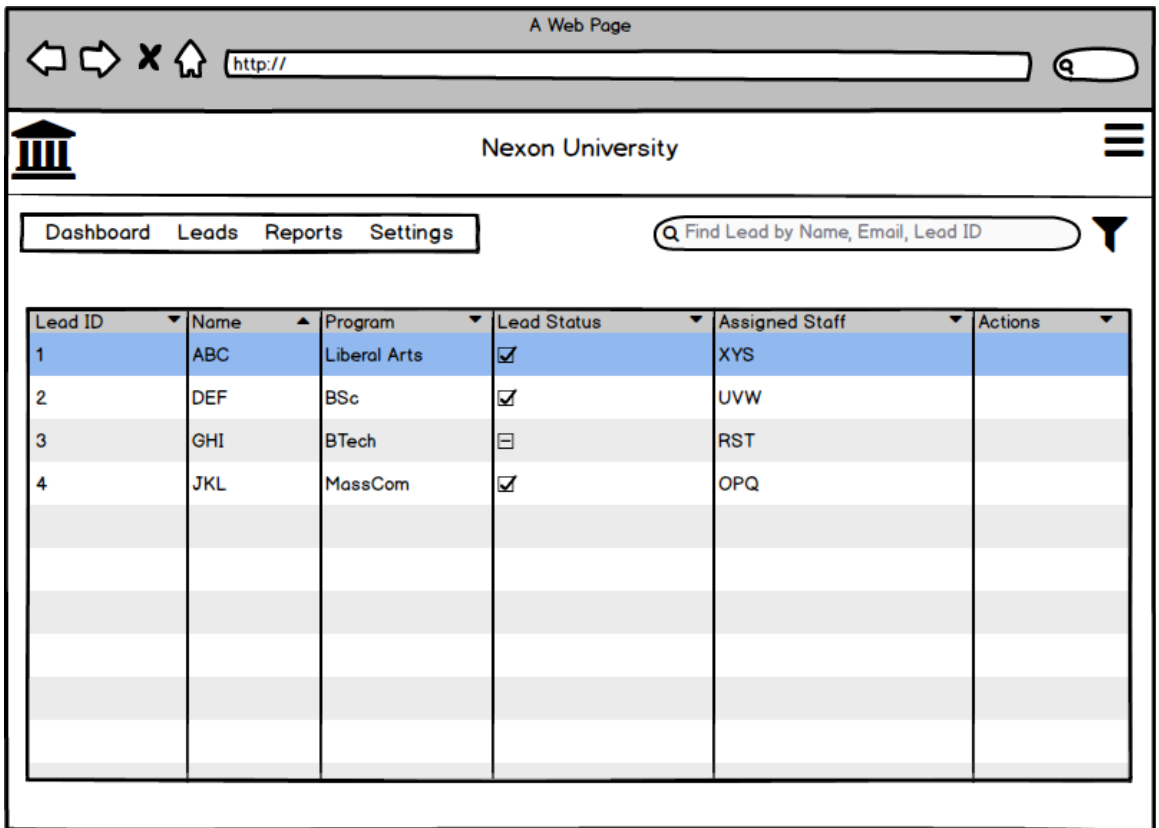
Use Case ID:	**UC010**
Use Case Name:	Application Verification
Created by:	Business Analyst
Use Case Description:	Verify the completeness and authenticity of submitted applications.
Actors:	● **Primary Actor:** Admissions Officer ● **Secondary Actor:** System
Basic Flow:	1. Admissions officer reviews submitted applications. 2. System flags incomplete or suspicious entries. 3. Officer marks applications as "verified" or "requires additional details."
Alternate Flow:	● **Auto Verification:** 1. System performs preliminary checks (e.g., mandatory fields, file formats).

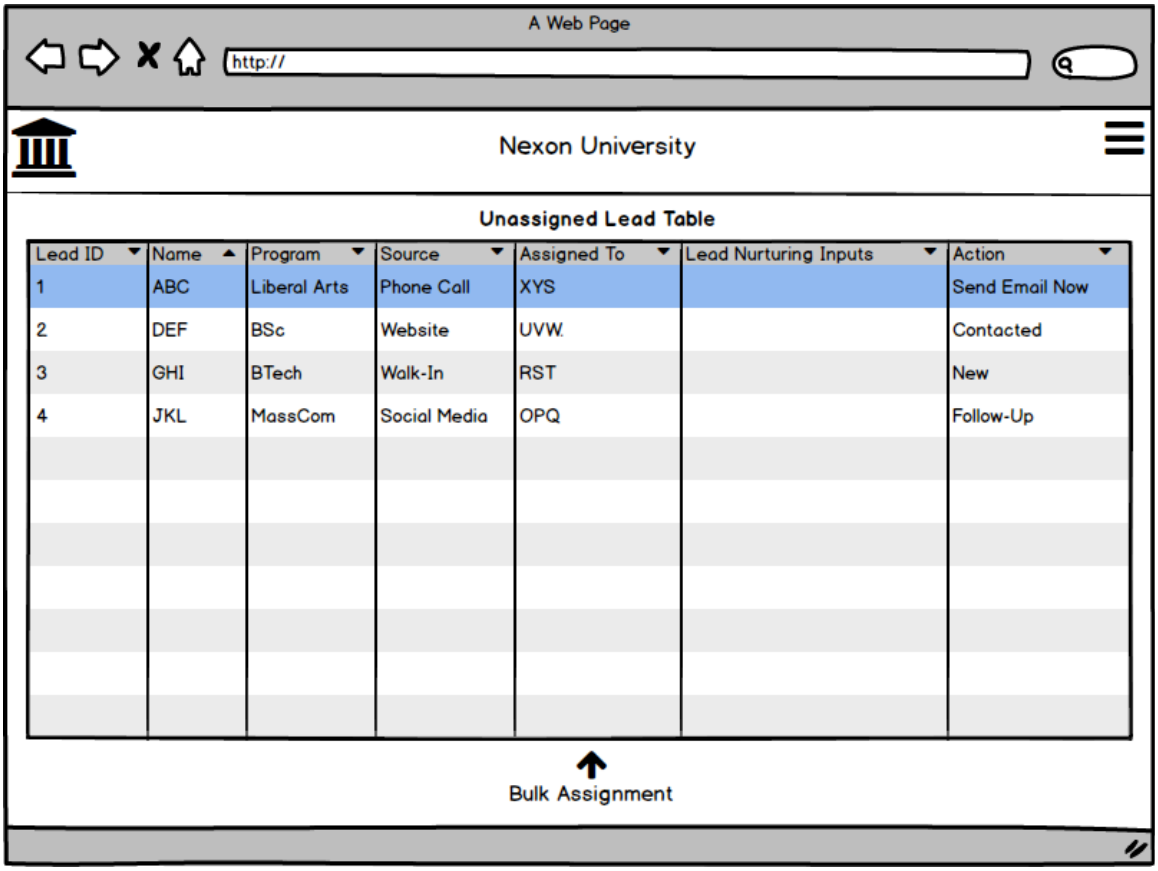
Exceptional Flow:	• **False Verification:** 1. Officer identifies and corrects errors flagged by the system.
Pre-conditions:	• Application must be submitted.
Post-conditions:	• Application verification status is updated.
Assumptions:	• Officers have access to all necessary tools and data.
Constraints:	• Verification process must be completed within 7 business days.
Dependencies:	• Applicant's supporting documents.
Inputs and Outputs:	• **Inputs:** Submitted application. • **Outputs:** Verification status.
Business Rules:	• Verified applications proceed to eligibility checks.
Miscellaneous Information:	• System maintains a log of verification actions.

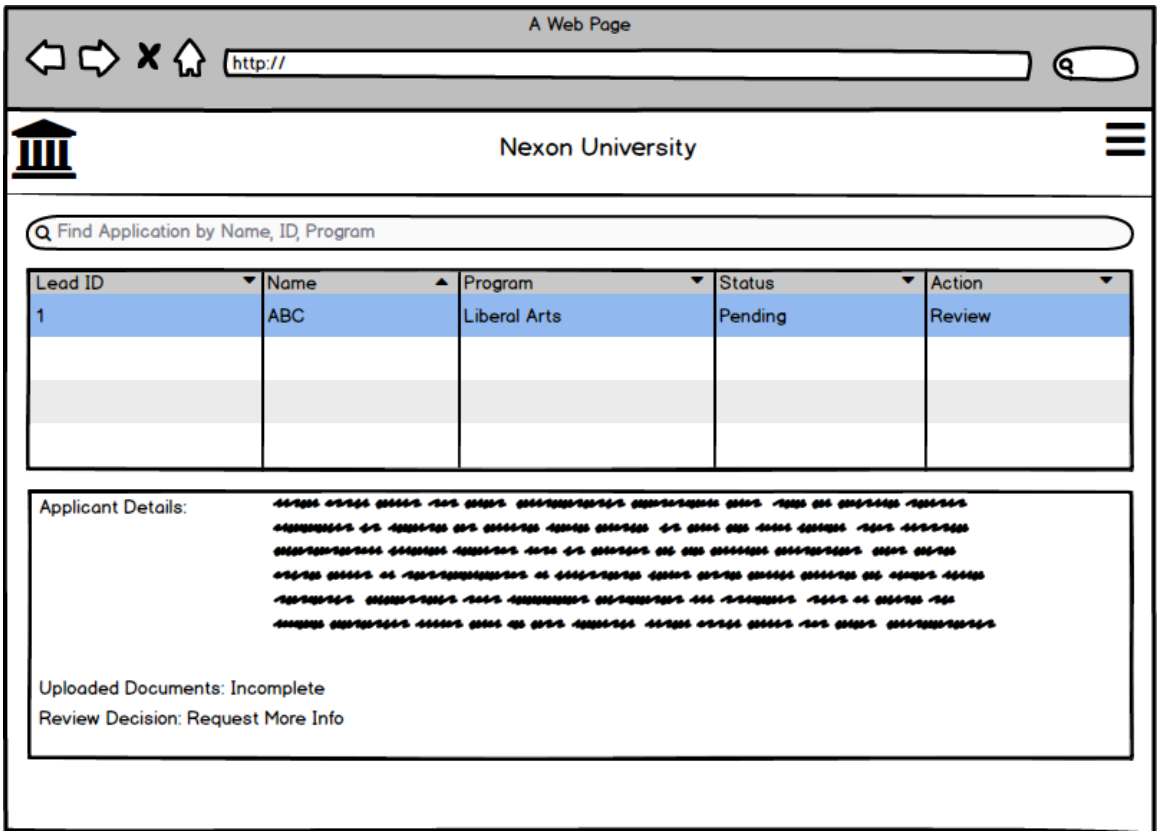
Document 7- Screens and pages




A Web Page
http://
Nexon University
Start Your Journey With Us..
Personal Details:
Name:
Email:
Phone:
Program Selection:
Upload Supporting Documents
I agree to receive communications
Submit


A Web Page
http://
Nexon University
Dashboard
Leads
Reports
Settings
Find Lead by Name, Email, Lead ID
Lead ID
Name
Program
Lead Status
Assigned Staff
Actions
1
ABC
Liberal Arts
XYS
2
DEF
BSc
UVW
3
GHI
BTech
RST
4
JKL
MassCom
OPQ


A Web Page
http://
Nexon University
Unassigned Lead Table
Lead ID
Name
Program
Source
Assigned To
Lead Nurturing Inputs
Action
1
ABC
Liberal Arts
Phone Call
XYS
Send Email Now
2
DEF
BSc
Website
UVW
Contacted
3
GHI
BTech
Walk-In
RST
New
4
JKL
MassCom
Social Media
OPQ
Follow-Up
Bulk Assignment


A Web Page
http://
Nexon University
Find Application by Name, ID, Program
Lead ID
Name
Program
Status
Action
1
ABC
Liberal Arts
Pending
Review
Applicant Details:
Uploaded Documents: Incomplete
Review Decision: Request More Info

A Web Page

http://

Nexon University

Find Application by Name, ID, Program

Lead ID	Name	Program	Contact	Status	Student ID	Action
1	ABC	Liberal Arts	9000190001	Converted to Student	123	Confirmation Email Drafting

Enrollment Confirmation Email Preview:

Save & Confirm

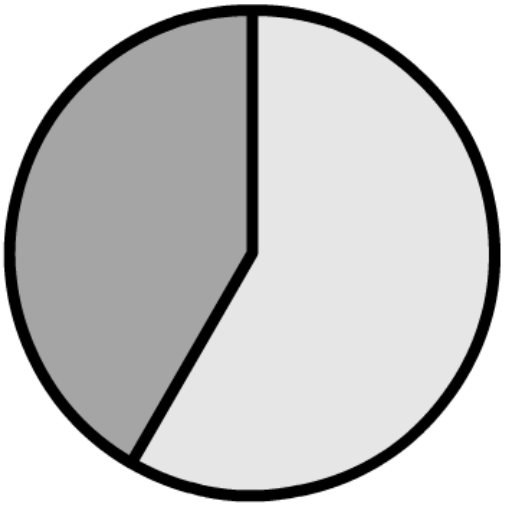
A Web Page

http://

Nexon University

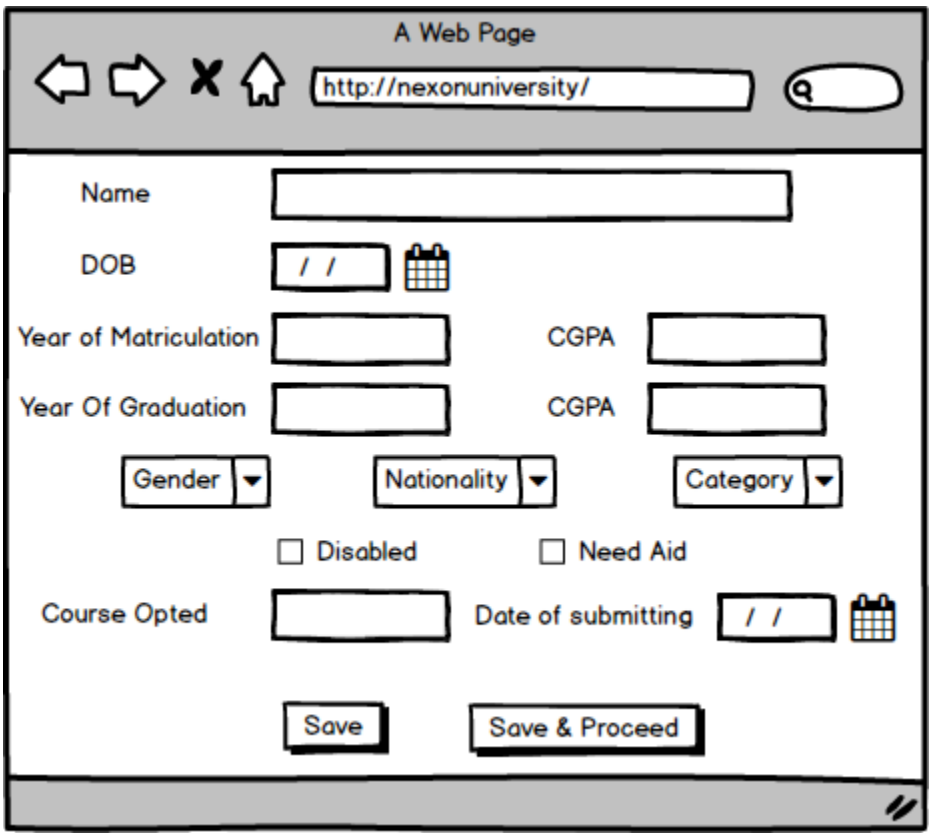
Total Leads | Conversion Rate | Top Sources | Application Status Breakdown

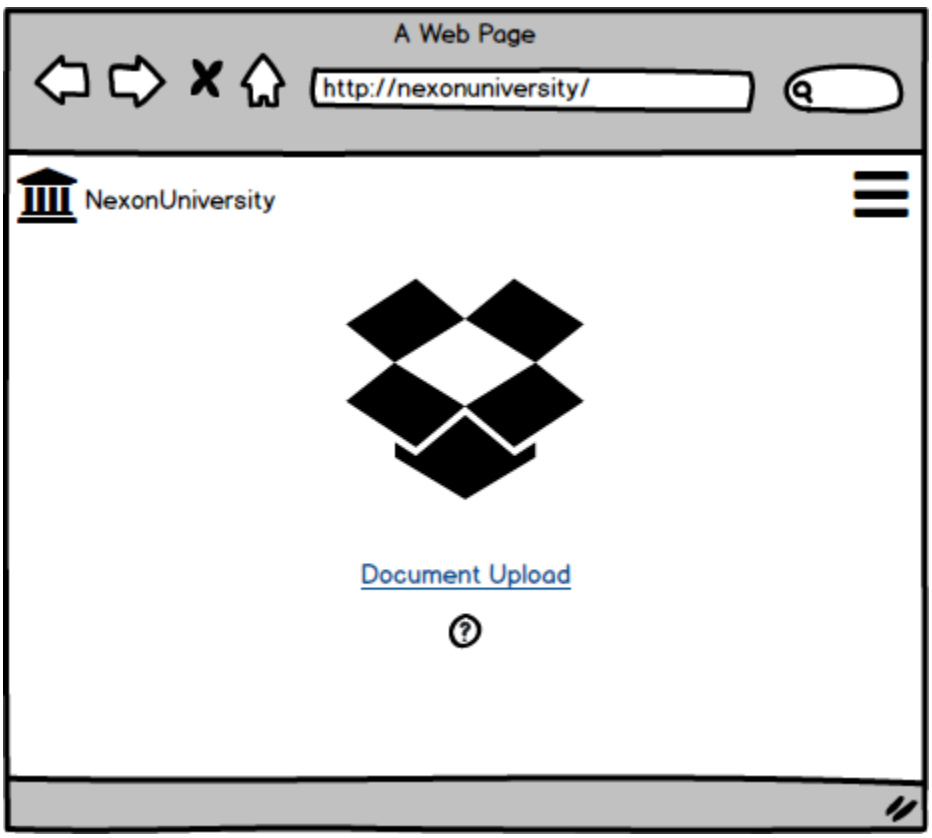
Filter by Date

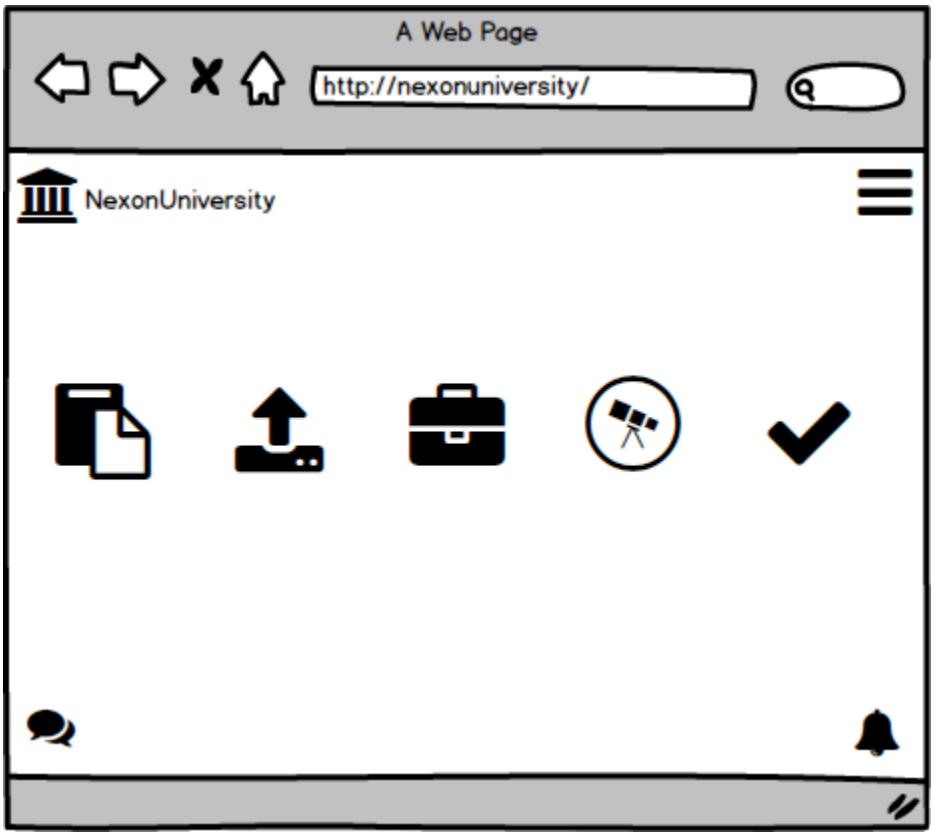


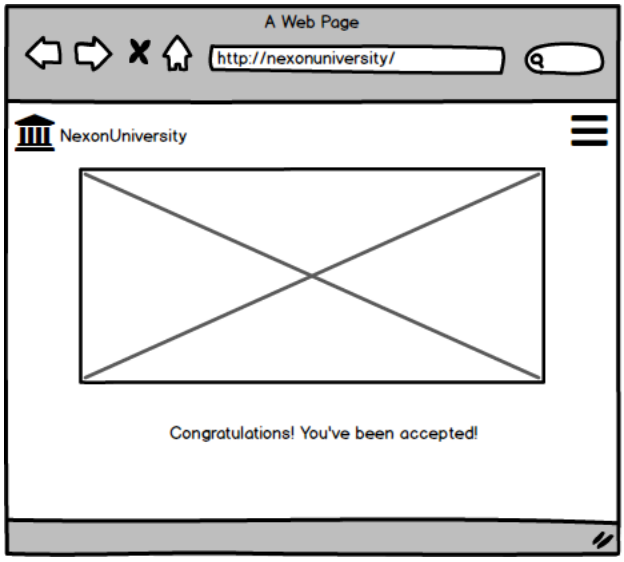
Download PDF

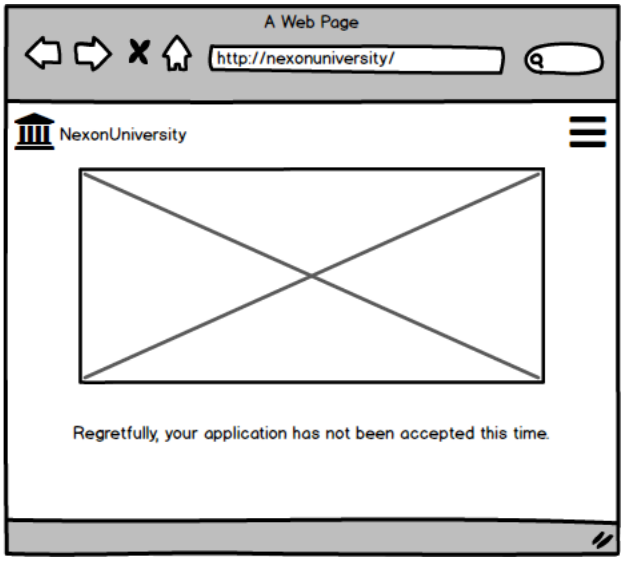
Export to Excel

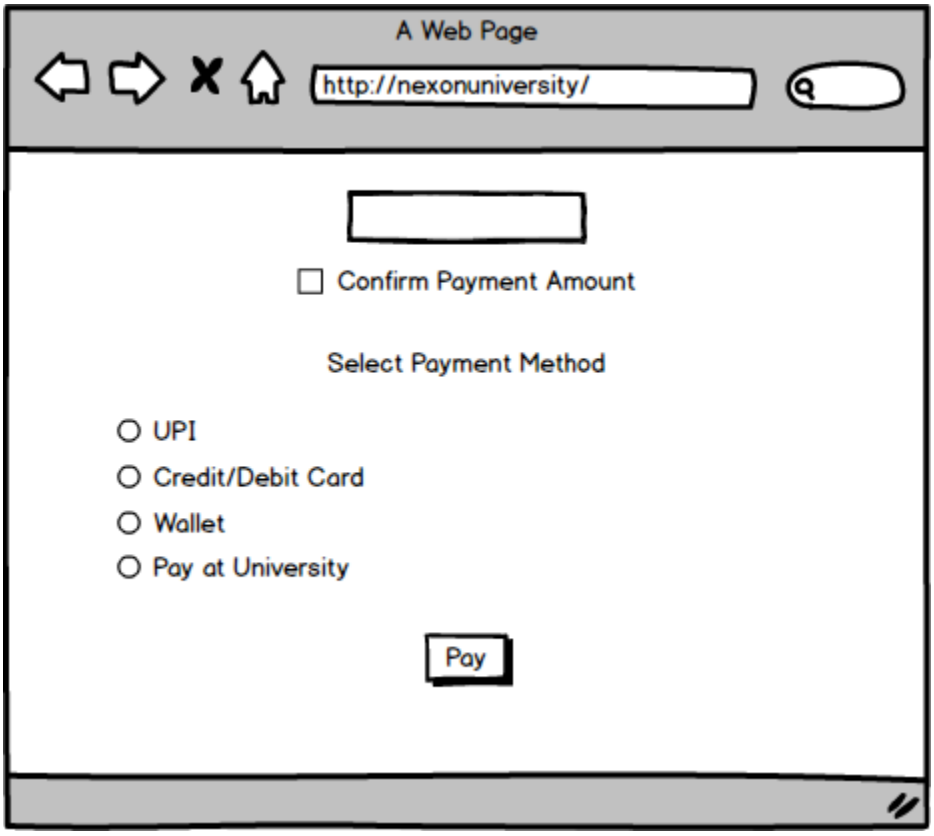

A Web Page
http://nexonuniversity/
Name
DOB
/ /
Year of Matriculation
CGPA
Year Of Graduation
CGPA
Gender
Nationality
Category
Disabled
Need Aid
Course Opted
Date of submitting
/ /
Save
Save & Proceed

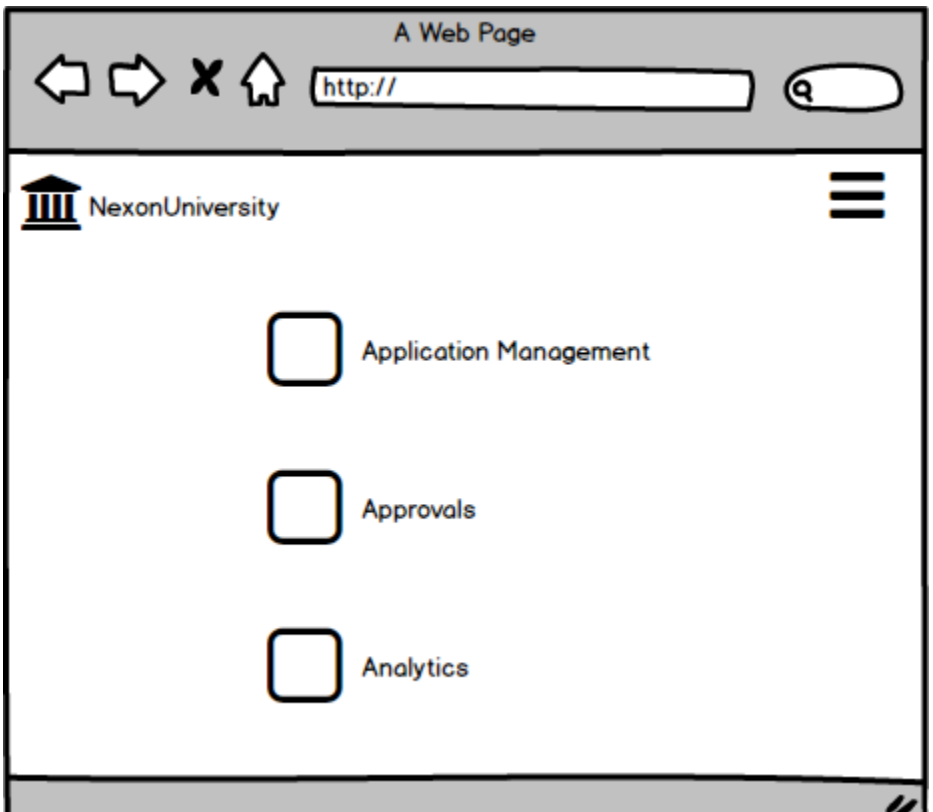

A Web Page
http://nexonuniversity/
NexonUniversity
Document Upload


A Web Page
http://nexonuniversity/
NexonUniversity


A Web Page
http://nexonuniversity/
NexonUniversity
Congratulations! You've been accepted!


A Web Page
http://nexonuniversity/
NexonUniversity
Regretfully, your application has not been accepted this time.


A Web Page
http://nexonuniversity/
Confirm Payment Amount
Select Payment Method
UPI
Credit/Debit Card
Wallet
Pay at University
Pay


A Web Page
http://
NexonUniversity
Application Management
Approvals
Analytics

Document 8- Tools-Visio and Balsamiq Write a paragraph on your experience using Visio and Balsamiq for the project.

Working on this project with Visio and Balsamiq was a great experience. Visio made it easy to map out use case diagrams, activity flows, and system interactions, helping me visualize how different users interact with the Graduate Admissions Tracking System Lead Management System (GATS LMS) and GATS. The structured connectors and drag-and-drop features saved a lot of time and kept everything organized. Balsamiq, on the other hand, was a game-changer for designing wireframes. Its simple, sketch-like style made it quick to create user-friendly layouts for key screens like the home page, lead capture, lead tracking & qualification, lead conversion page, application submission, document upload, tracking pages, and, report & analytics page. What I loved most was how both tools worked together—Visio helped define the system's logic, while Balsamiq brought the UI concepts to life. It was a smooth process that made designing this system both structured and creative.

Document 9- BA experience My experience as BA in following phases:

1. Requirement Gathering
   - We used the MoSCoW technique to categorize requirements into Must-haves, Should-haves, Could-haves, and Won't-haves, ensuring prioritization.
   - Since the client was unavailable at times, I identified and coordinated with alternative stakeholders to gather necessary information without delays.
   - Applied the FURPS (Functionality, Usability, Reliability, Performance, Supportability) model to validate requirements and ensure completeness.
   - Identified and removed redundant and duplicate requirements, streamlining the project scope.
   - Used prototyping techniques to provide the client with a visual representation of system functionalities, leading to more specific and refined requirements.
   - Conducted stakeholder interviews and workshops to capture both functional and non-functional requirements for LMS and GATS.

2. Requirement Analysis
   - Created UML diagrams (Use Case Diagrams, Class Diagrams, and Sequence Diagrams) to visually represent system interactions and workflows.
   - Developed Activity Diagrams to outline the step-by-step process flow of key functionalities such as lead tracking, application processing, and document verification.
   - Communicated these diagrams with the development and testing teams; incorporated feedback and modifications to ensure clarity and feasibility.

- Prepared key documentation, including the Business Requirements Specification (BRS) and System Requirements Specification (SRS), ensuring alignment between business needs and technical implementation.
- Worked on data flow diagrams (DFD) for LMS and GATS to describe how information moves between different system components.

3. Design
- Derived test cases from use case diagrams, ensuring comprehensive test coverage.
- Regularly communicated with the client and design team to refine system UI/UX through iterative discussions.
- Created both positive and negative test cases, ensuring edge cases were handled effectively.
- Ensured that no test case was overlooked, preventing potential issues in the later development stages.
- Generated and maintained test data for different application scenarios, including document uploads, payment processing, and status updates.
- Continuously updated the Requirement Traceability Matrix (RTM) to ensure every requirement had corresponding test cases.

4. Development
- Organized and led Joint Application Development (JAD) sessions with key stakeholders to finalize implementation strategies.
- Acted as a bridge between the business and technical teams, resolving queries raised by developers during implementation.
- Addressed resistance from some team members regarding certain requirements by conducting one-on-one discussions, explaining the impact of their concerns on the overall project.
- Referred to UML diagrams and workflows to guide development efforts and ensure accurate coding.
- Conducted regular stand-up meetings with developers and clients to track progress, clarify doubts, and resolve blockers.
- Ensured meeting recordings were made available for absent stakeholders, followed by individual discussions for better alignment.

5. Testing
- Developed detailed test cases based on use cases, ensuring full functionality testing of lead capture, admission tracking, and notifications.
- Conducted high-level testing to validate system stability before QA testing.
- Requested and coordinated test data from the client for real-world scenario testing.
- Updated the RTM, ensuring all requirements had been properly tested and validated.
- Obtained formal sign-off from the client, confirming that all business expectations were met.
- Prepared the client for User Acceptance Testing (UAT) by conducting training sessions and clarifying any last-minute queries.

6. Deployment

- Shared the finalized RTM with the client as part of the project closure document.
- Coordinated with stakeholders to complete and distribute end-user manuals for seamless adoption.
- Organized and scheduled training sessions for users, ensuring smooth onboarding and reducing post-deployment issues.
- Ensured all relevant stakeholders attended the training sessions and had clarity on system usage and troubleshooting.
- Monitored the system post-deployment for any immediate bug fixes or enhancement requests.

Working on the GATS LMS and Graduate Admissions Tracking System (GATS) project as a Business Analyst was an enriching experience. The project involved detailed requirement gathering, constant collaboration with stakeholders, and proactive problem-solving. From requirement validation to deployment, I ensured the system met all business and user expectations while maintaining efficiency and clarity throughout the SDLC. This experience reinforced my ability to manage complex business needs, stakeholder expectations, and technical feasibility seamlessly.